Artificial Intelligence and Computing Innovations

Intelligence Science Publishing

https://www.ispublishing.com

ISSN: 3070-4944

Vo.4, No.1, 2024

Automatic Orchestration of Enterprise ETL Processes via Graph Neural Network and Large Language Model Collaborative Reasoning

Ye Zhang[1]*, Steven Lu[2]

[1]*Cornell Tech, New York, USA*

[2]*University of Cincinnati, Cincinnati, USA*

**Corresponding author: yz2778@cornell.edu*

Abstract: As data source types, field semantics, quality rules, and task dependencies in enterprise data platforms become increasingly complex, traditional ETL process orchestration methods relying on manual configuration and static rules struggle to meet the demands for automation, consistency, and reliability in large-scale enterprise data processing. Addressing the limitations of traditional ETL process orchestration methods, which suffer from insufficient cross-source relationship identification, inadequate business semantic understanding, and the ever-increasing complexity of contract types, field semantics, quality rules, and task dependencies, this paper proposes an automatic orchestration method based on collaborative reasoning using graph neural networks and a large language model. This method first models data sources, fields, rules, and tasks as a unified ETL state graph and uses relation-aware graph representation learning to capture high-order relationships between field mappings, task dependencies, and rule constraints. Subsequently, it constructs a structural semantic joint context by combining business requirement text, data pattern information, and rule descriptions, enabling the large language model to complete task decomposition, process reasoning, and candidate ETL process generation under graph-structured evidence constraints. Furthermore, the method introduces constraint consistency verification and a multi-criteria scoring mechanism to screen candidate processes based on dependency rationality, rule coverage, and execution feasibility, thereby improving the stability and reliability of the automatic orchestration results. Comparative experiments based on publicly available enterprise data integration scenarios show that the proposed method has good overall performance in terms of process generation correctness, dependency consistency, rule coverage, and execution success rate, and can provide effective support for enterprise data warehouse construction, automated data governance, and intelligent data engineering.

Keywords: Enterprise-level ETL; Graph Neural Networks; Large Language Models; Automatic Orchestration

1. Introduction

As enterprises continue their digital transformation and data-driven decision-making, the scale of data flow between business systems, data warehouses, log platforms, external interfaces, and cloud services is constantly expanding. Enterprise-level data governance is gradually shifting from single data processing tasks to the collaborative management of complex data processes [1,2]. ETL processes, as a crucial link connecting data sources and data applications, undertake core functions such as data extraction, cleaning, transformation, verification, and loading. Their execution quality directly impacts the reliability of data warehouse construction, business analysis, risk monitoring, and intelligent decision-making. In multi-source heterogeneous data environments, differences in data structures, inconsistent field semantics, frequent changes in business rules, and complex upstream and downstream dependencies make traditional ETL development models, which rely on manual configuration and static rules, difficult to meet enterprises' real-world needs for high efficiency, high consistency, and high maintainability [3].

Existing enterprise-level ETL processes typically require engineers to manually understand data relationships, design transformation logic, configure scheduling dependencies, and handle exceptions based on business requirements. This approach is applicable to small-scale or stable data scenarios, but in large-scale enterprise data platforms, it easily exposes problems such as long process orchestration cycles, difficulty in identifying cross-table relationships, high task dependency maintenance costs, and delayed exception handling. Especially with the continuous iteration of business systems, dynamic changes in data patterns, and ongoing updates to data quality constraints, ETL processes are no longer simple linear processing chains, but complex task networks composed of data entities, field semantics, transformation rules, quality constraints, and scheduling relationships [4]. Therefore, understanding the structural dependencies and semantic relationships within an enterprise data pipeline from a global perspective, and achieving automated process orchestration based on this understanding, has become a crucial issue for improving the intelligence level of enterprise data engineering.

Graph neural networks have strong advantages in relational modeling and structural representation learning. They can uniformly represent enterprise data sources, field nodes, transformation relationships, quality rules, and task dependencies as graph structures, thereby capturing local connectivity and global topological features in the ETL process. Compared to traditional rule-based matching or template retrieval methods, graph structure modeling can more fully express the strength of associations, dependency paths, and constraint propagation relationships between data objects, providing a structured foundation for complex process planning. Meanwhile, large language models demonstrate strong capabilities in natural language understanding, business semantic parsing, task decomposition, and logical reasoning, capable of transforming business requirements, data descriptions, and rule descriptions into executable processing intentions [5]. Combining the structure-aware capabilities of graph neural networks with the semantic reasoning capabilities of large language models helps overcome the limitations of single models in enterprise ETL automatic orchestration, enabling the process generation process to simultaneously possess relational understanding, semantic alignment, and logical planning capabilities.

Research on enterprise-level ETL process automatic orchestration based on collaborative reasoning of graph neural networks and large language models has significant theoretical and practical value. From a theoretical perspective, this direction helps to promote the transformation of data engineering tasks from experience-driven and rule-driven to joint structure-semantic modeling, further expanding the boundaries of the integrated application of graph learning and large language models in the automation of complex business processes. From an application perspective, automatic orchestration mechanisms can reduce the development and maintenance costs of enterprise data processes, improve the consistency, traceability, and adaptability of cross-source data processing, and provide a more stable data foundation for data warehouse construction, data governance, business intelligent analysis, and enterprise-level intelligent decision-making. Therefore, building automatic orchestration methods with both structure modeling and semantic reasoning capabilities for enterprise-level ETL scenarios is of practical significance for improving the intelligence, automation, and trustworthiness of modern enterprise data platforms.

2. BackGround

The construction of enterprise data platforms is gradually shifting from a stage centered on data aggregation to one centered on intelligent processes, semantic consistency, and automated governance. In this process, ETL is no longer merely responsible for data transport and format conversion, but is gradually becoming a crucial infrastructure for implementing enterprise business rules, controlling data quality, and facilitating cross-system collaboration [6]. As the number of internal enterprise application systems increases and data source types expand, the relationships between data tables, the semantic mapping between fields, and the execution dependencies between tasks become increasingly complex. Traditional ETL processes often rely on manual understanding of business documents, writing transformation scripts, and configuring scheduling rules. While these can complete data processing tasks in specific scenarios, they are prone to problems such as insufficient process reusability, low orchestration efficiency, and high maintenance difficulty when facing dynamically changing data patterns and continuously evolving business needs.

In the context of intelligent data engineering, automated ETL orchestration is gradually becoming an important direction for improving enterprise data processing capabilities. This task not only requires the system to identify the structural relationships between data sources but also to understand the inherent connections between business semantics, data constraints, and processing objectives. Graph neural networks can express the connections between complex data objects through graph structures, providing structured support for process dependency analysis and transformation path modeling; large language models can perform semantic parsing and reasoning on natural language requirements, field descriptions, and business rules, providing semantic understanding capabilities for task decomposition and process generation [7]. The synergistic combination of the two enables ETL orchestration to shift from simple rule matching to an intelligent reasoning process driven by both structural relationships and business semantics, thus providing a new technological foundation for the automatic generation, dynamic adjustment, and reliable execution of enterprise data processes.

3. Methodology

The key to enterprise-level ETL automatic orchestration does not lie in directly translating business requirements into fixed scripts, but in constructing a unified and inferable task space from multi-source data objects, field semantics, quality constraints, and scheduling dependencies. For complex enterprise data environments, this study abstracts data tables, fields, transformation operators, quality rules, and execution tasks into a node set, while field mappings, foreign-key dependencies, transformation calls, rule constraints, and task precedence relations are abstracted into an edge set, thereby forming a heterogeneous state graph capable of simultaneously representing data structures and processing logic. This paper presents the overall architecture diagram of the algorithm, as shown in Figure 1.

To prevent the orchestration process from relying only on shallow matching of table names or field names, the initial node representation is jointly composed of structural attributes, semantic descriptions, data quality states, and historical execution features, enabling the model to characterize the business meaning, constraint conditions, and operational context of data objects within a unified representation space. The enterprise-level ETL state graph is defined as:

$$G_{etl}=(V,\mathcal{E},\boldsymbol{X},\mathcal{R}) \tag{1}$$

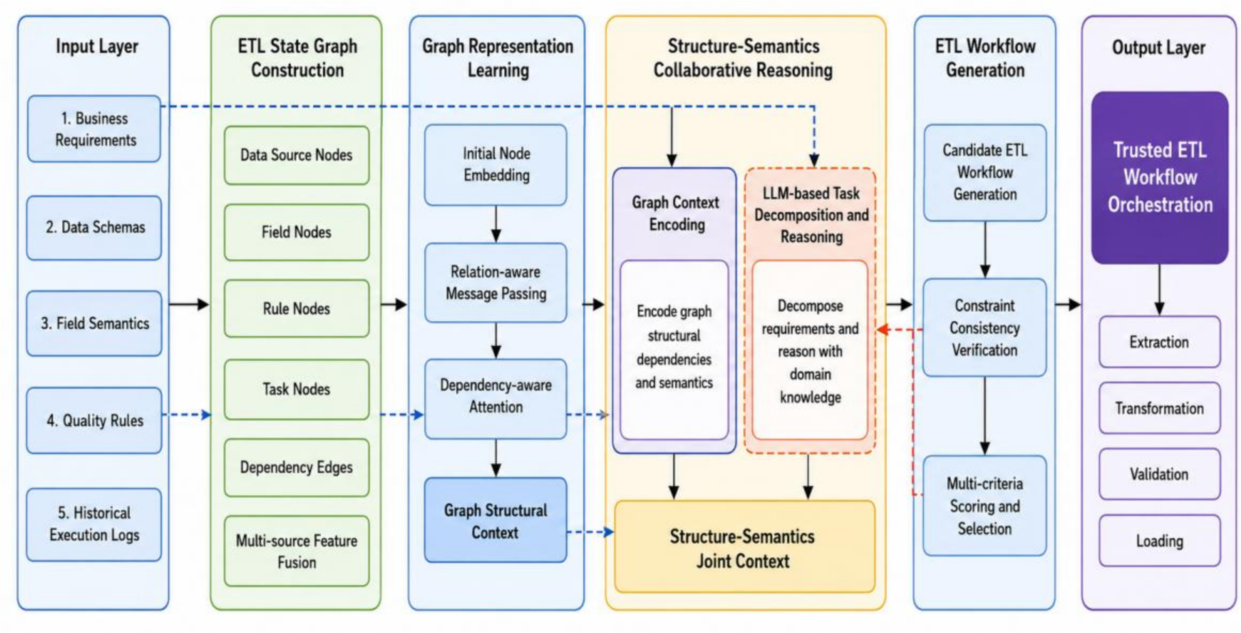

Figure 1. Overall model architecture diagram

The graph structure G_{etl} carries the overall state of the enterprise data workflow, while the node set V covers objects such as data sources, fields, rules, and tasks; the edge set $\mathscr{E}$ describes the dependency relationships among these objects; and the node feature matrix $\boldsymbol{X}$ together with the relation type set $\mathscr{R}$ represents node attributes and multi-category workflow constraints, respectively. To enhance field-level semantic alignment, the node representation incorporates not only structural information from the database schema, but also semantic embeddings generated from business documents, field annotations, and transformation descriptions, allowing differentiated names of the same business concept across different systems to be mapped into a more stable latent space. After integrating the structural feature $\mathbf{x}_i^s$, semantic feature $\mathbf{x}_i^m$, and quality state feature $\mathbf{x}_i^q$, the initial representation of the i-th node is constructed as:

$$\boldsymbol{h}_i^0=\phi\left(\boldsymbol{W}_s\boldsymbol{x}_i^s+\boldsymbol{W}_m\boldsymbol{x}_i^m+\boldsymbol{W}_q\boldsymbol{x}_i^q+\boldsymbol{b}\right) \tag{2}$$

This representation transforms ETL orchestration from isolated script generation into a global understanding process centered on data-object relationships, in which the mapping matrices $\boldsymbol{W}_s$, $\boldsymbol{W}_m$, and $\boldsymbol{W}_q$ correspond to learnable projections of features from different sources, and the nonlinear function $\phi(\cdot)$ is used to alleviate scale differences among heterogeneous features and form a unified node state. The initial graph representation obtained in this way provides the basis for subsequent relational propagation, enabling field similarity, constraint transmission, and task dependency to be jointly modeled within a unified graph structure.

The objective of the relational propagation stage is to extract high-order dependencies from local data connections for workflow planning, rather than merely determining whether a direct association exists between two fields or two tasks. Considering the large number of relation types, strong directionality, and long constraint propagation paths in enterprise ETL, this study adopts a relation-aware graph message-passing mechanism, in which different edge types are assigned independent transformation parameters during information aggregation, thereby avoiding the mixture of field mapping, task scheduling, and quality constraints into a single relation pattern. For node i, its representation at the $(l+1)$-th layer is jointly determined by neighboring nodes, relation types, and its own state, which is formulated as:

$$\boldsymbol{h}_i^{l+1}=\sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in N_i^r}\alpha_{ij}^{r,l}\,\boldsymbol{W}_r^l\boldsymbol{h}_j^l+\boldsymbol{W}_0^l\boldsymbol{h}_i^l\right) \tag{3}$$

Here, N_i^r denotes the set of neighbors connected to node i under relation r, the weight $\alpha_{ij}^{r,l}$ describes the structural contribution of neighbor node j to node i during propagation at layer l, and the relation projection matrix $\boldsymbol{W}_r^l$ preserves the differences among edge types in workflow reasoning. Considering that not all dependencies in enterprise data pipelines have the same reliability, the attention weight is further computed from the source node, target node, and relation embedding, so that critical dependency paths can receive stronger responses during propagation. The relation attention calculation is expressed as:

$$\alpha_{ij}^{r,l}=\frac{exp\left(\boldsymbol{a}_r^{\top}\ [\boldsymbol{W}_r^l\boldsymbol{h}_i^l\|\boldsymbol{W}_r^l\boldsymbol{h}_j^l\|\boldsymbol{e}_r]\right)}{\sum_{k\in N_i^r}exp\left(\boldsymbol{a}_r^{\top}\ [\boldsymbol{W}_r^l\boldsymbol{h}_i^l\|\boldsymbol{W}_r^l\boldsymbol{h}_k^l\|\boldsymbol{e}_r]\right)} \tag{4}$$

This design enables the model to prioritize structural evidence that is more critical to workflow generation in complex dependency networks, such as primary-foreign key constraints, field transformation paths, and upstream task states, thereby improving the structural awareness of automatic orchestration for enterprise data relationships. After multi-layer propagation, the graph representation is no longer limited to the original data schema, but instead integrates high-order information such as cross-source field correspondence, constraint conflict propagation, and task-chain dependencies, providing more stable structural context for subsequent large language model reasoning.

The large language model undertakes semantic understanding and workflow reasoning in the proposed method, but its generation process is not completed independently from the graph structure; instead, it is constrained by the structural evidence output by the graph neural network. Since natural language business requirements may suffer from incomplete descriptions, ambiguous field references, and unclear rule boundaries, this study organizes user requirements, data dictionaries, quality rules, and graph-structure summaries into a reasoning context, enabling the language model to perform task decomposition based on the real dependency relationships among enterprise data objects. Given the business requirement text Q and graph representation $\boldsymbol{H}^L$, the structural-semantic joint context is represented as:

$$\boldsymbol{z}=\psi\left(Pool(\boldsymbol{H}^L),Enc(Q),Enc(C)\right) \tag{5}$$

The context set C contains auxiliary information such as field descriptions, transformation rules, and quality constraints; the encoding function $\mathrm{Enc}(\cdot)$ maps textual semantics into vector representations; and the aggregation function $\mathrm{Pool}(\cdot)$ extracts a workflow-level structural summary from the final graph representation $\boldsymbol{H}^L$. Unlike ETL script generation methods that rely only on prompts, the joint context $\boldsymbol{z}$ explicitly binds business intent with structural dependencies, allowing the language model to simultaneously consider whether data sources match, whether field mappings are reasonable, whether transformation orders satisfy dependencies, and whether quality rules are covered during workflow generation. Based on this context, the automatic orchestration process is modeled as a conditional generation problem of candidate workflow sequences, with the objective defined as:

$$P(P\mid Q,G_{etl})=\prod_{t=1}^{T}P\left(p_t\mid p_{<t},\boldsymbol{z}\right) \tag{6}$$

In this formulation, the workflow sequence $\mathrm{P}=\{p_1,p_2,\dots,p_T\}$ consists of operation units such as extraction, cleaning, transformation, validation, and loading, while the generation of the current step p_t is jointly constrained by the historical steps $p_{<t}$ and the structural-semantic context $\boldsymbol{z}$. Through this collaborative reasoning mechanism, the language model generates not isolated textual workflow descriptions, but candidate execution plans consistent with the enterprise data graph structure, thereby enhancing the interpretability and controllability of ETL automatic orchestration.

After workflow generation, the candidate orchestration results still need consistency verification and trustworthy selection, because incorrect execution orders, omitted quality rules, or transformation paths that fail to satisfy dependency constraints may lead to systematic bias in downstream data warehouses in enterprise-level ETL scenarios. Therefore, this study introduces structural constraint scoring, semantic matching scoring, and execution feasibility scoring to jointly evaluate the quality of candidate workflows, so that the final selected result can satisfy data dependencies, business semantics, and engineering execution requirements at the same time. Assuming that the overall score of a candidate workflow P is jointly determined by three types of constraints, the trustworthy orchestration objective is formulated as:

$$\mathrm{S(P)}=\lambda_g \mathrm{S}_g(\mathrm{P,G}_{etl})+\lambda_m \mathrm{S}_m(\mathrm{P,Q})+\lambda_c \mathrm{S}_c(\mathrm{P,C}) \tag{7}$$

The weights λ_g, λ_m, and λ_c control the influence of structural dependencies, business semantics, and rule constraints in workflow selection, respectively, allowing different enterprise scenarios to adjust orchestration preferences according to governance objectives. The structural constraint score emphasizes the consistency between workflow steps and dependency paths in the graph, the semantic matching score focuses on whether the generated operations cover business requirements, and the rule constraint score is used to determine whether key requirements such as data quality checks, primary-key uniqueness, referential integrity, and loading conditions are explicitly incorporated into the workflow. The final orchestration result is determined by the workflow with the highest score in the candidate set Ω, and the selection process is expressed as:

$$\mathrm{P}^* = \underset{\mathrm{P}\in\Omega}{\arg\max}\, \mathrm{S(P)} \tag{8}$$

Compared with directly generating a single workflow, this optimization form enables constraint-aware selection among multiple candidate solutions, thereby reducing the impact of uncertainty in language model reasoning on the reliability of enterprise data workflows. Overall, the proposed method provides structural dependency modeling capability through graph neural networks, semantic reasoning and task planning capability through large language models, and constraint-based screening of candidate workflows through a trustworthy scoring mechanism, allowing enterprise-level ETL automatic orchestration to form a closed loop across data relationship understanding, business intent parsing, and workflow consistency verification.

4. Experimental Results and Analysis

4.1 Dataset

The TPC-DI public dataset is selected as the foundational data source for enterprise-level ETL process auto-orchestration tasks. This dataset is geared towards data integration and data warehouse construction scenarios, encompassing source data from transaction systems and other heterogeneous sources. Through extraction, transformation, and loading processes, multi-source data is organized into a unified data warehouse model. Its data model is based on enterprise brokerage business, including various source data modes, file formats, target data warehouse modes, transformation rules, historical loading tasks, and incremental update tasks. It can effectively simulate common cross-source data fusion, field mapping, quality verification, dependency constraints, and process execution issues in enterprise ETL scenarios. Compared to datasets focused solely on single-table analysis or static queries, TPC-DI emphasizes the complete data flow from the source to the warehouse, making it suitable for constructing ETL state diagrams, modeling task dependencies, and supporting research on auto-orchestration methods based on collaborative reasoning using graph neural networks and large language models.

4.2 Experimental setup

To ensure the reproducibility and fairness of the enterprise-level ETL process automatic orchestration experiment, the experiment was conducted in a unified data processing, model training, and process generation environment. The dataset used was the TPC-DI open-source dataset, and an ETL state graph was constructed based on the source data pattern, field semantics, target repository structure, transformation rules, and task dependencies. During the experiment, the data source table, field descriptions, quality constraints, and loading rules were first standardized and parsed, and mapped to graph nodes and dependency edges. Then, a graph neural network was used to learn the structured process context, which, along with the business requirement text, was input into a large language model to complete task decomposition and candidate process generation. To avoid additional impacts from different model configurations on the results, all methods adopted the same data partitioning, preprocessing strategy, and evaluation process. The training set, validation set, and test set were divided in a 7:1:2 ratio, and key parameters such as optimizer, learning rate, batch size, and number of training epochs remained consistent. Specific experimental environment and parameter settings are shown in Table 1.

Table 1. Experimental Environment and Parameter Settings

Configuration Item	Specific Setting
Programming Language	Python 3.10
Deep Learning Framework	PyTorch 2.1
Number of Graph Neural Network Layers	3 layers
Hidden Dimension	256
Batch Size	32
Training Epochs	50
Optimizer	AdamW
Initial Learning Rate	0.0001
Weight Decay	0.0001
Dropout	0.2
Large Language Model	Qwen-7B-Chat
Maximum Generation Length	1024
Decoding Temperature	0.2
Computing Device	NVIDIA H100 GPU

4.3 Experimental Results and Analysis

To evaluate the comprehensive applicability of the proposed method in enterprise-level ETL automated orchestration tasks, representative methods in related areas such as data preparation, data cleaning, data transformation, process debugging, and language model-assisted data processing were selected for comparison. These methods cover rule-driven, interactive transformation, feedback optimization, reinforcement learning, error correction, and language model data processing techniques, reflecting the differences between various automated orchestration mechanisms in terms of process generation, dependency consistency, rule coverage, and execution feasibility. The experimental results are shown in Table 2. This paper uses four metrics—FGA, DCS, RCR, and ESR—to measure the candidate process generation quality, structural dependency consistency, quality rule coverage, and end-to-end execution success rate. All evaluation metrics are expressed as percentages.

Table 2. Comparative results of related methods and the proposed method

Method	FGA	DCS	RCR	ESR
Giovanelli et al. [8]	75.38	73.41	70.62	71.85
Konstantinou et al. [9]	77.64	74.29	72.35	73.68
Berti-Équille et al. [10]	79.45	78.16	76.29	75.92
Kandel et al. [11]	82.37	80.42	78.96	79.55
Lourenço et al. [12]	80.86	82.74	77.83	81.29
Mahdavi et al. [13]	83.25	81.38	84.17	80.64
Jaimovitch-López et al. [14]	85.92	83.67	82.51	84.36
Ours	93.58	94.27	92.84	95.16

FGA measures the consistency between the automatically generated ETL process and the target business requirements, focusing on whether the process steps can fully cover core tasks such as data extraction, cleaning, transformation, and loading. DCS evaluates the rationality of dependencies between data objects, field mappings, and task nodes in the generated process, reflecting the model's understanding of the enterprise's data link structure. RCR mainly characterizes the coverage of quality rules, business constraints, and integrity verification conditions in the generated process, reflecting the method's responsiveness to data governance requirements. ESR measures the feasibility of the generated process at the actual execution level, focusing on whether the process can complete in the expected order and maintain good engineering stability.

The results in the table show that the proposed algorithm exhibits stronger comprehensive capabilities in process generation, dependency modeling, rule coverage, and execution feasibility. This method uses graph neural networks to characterize structural dependencies in enterprise ETL scenarios and leverages large language models to enhance business semantic understanding and task decomposition capabilities, enabling the generated process to not only match business requirements well but also maintain high constraint consistency and execution reliability. Compared to methods that rely solely on rules, interactive configuration, or localized process optimization, the method presented in this paper can simultaneously handle the relationships between data structures, field semantics, quality rules, and task dependencies within a unified framework, thereby generating more stable, complete, and reliable automatic orchestration results.

To further verify the roles of each core component in enterprise-level ETL automatic orchestration tasks, ablation experiments were conducted focusing on the three modules of the proposed method: relation-aware graph representation learning, structural semantic collaborative reasoning, and constraint consistency verification. While maintaining consistency in data sources, training strategies, and evaluation processes, corresponding modules were removed, and changes in process generation accuracy, dependency consistency, rule coverage, and execution success rate were observed to analyze the impact of different mechanisms on automatic orchestration quality. Specific ablation results are shown in Table 3.

Ablation experiments show that the proposed algorithm exhibits good overall performance across all evaluation metrics, indicating that relation-aware graph representation learning, structural semantic collaborative reasoning, and constraint consistency verification all play crucial roles in the automatic orchestration of enterprise-level ETL processes. Removing relation-aware graph representation learning negatively impacts the model's ability to characterize field mapping, task dependencies, and rule propagation relationships, leading to a decline in the generated process's structural consistency and process completeness. Removing structural semantic collaborative reasoning weakens the synergistic relationship between business requirement understanding and graph structure information, making it difficult for the process generation process to fully integrate data dependencies and semantic intent, thus significantly affecting the overall orchestration quality. Removing constraint consistency verification, while the generated process still maintains some task organization capabilities, insufficient verification of quality rules, execution order, and dependency constraints affects the process's reliability and executability. The complete algorithm simultaneously utilizes graph structure modeling, language reasoning, and constraint filtering mechanisms, enabling the generated ETL process to achieve more comprehensive synergistic support in terms of requirement coverage, dependency consistency, rule response, and execution stability.

Table 3. Ablation Study Results

Ablation Setting	FGA	DCS	RCR	ESR
w/o Relation-aware Graph Representation Learning	88.62	87.95	86.73	88.04
w/o Structure-Semantics Collaborative Reasoning	86.91	87.24	85.68	86.75
w/o Constraint Consistency Verification	90.17	89.63	86.42	88.96
Ours	93.58	94.27	92.84	95.16

The number of layers in a graph neural network is a crucial structural parameter affecting the state graph representation capability of enterprise-level ETL. Its setting directly relates to the propagation range of field mappings, task dependencies, and rule constraints within the graph. A graph neural network that is too shallow struggles to fully capture high-order dependencies across data sources and task chains, while an overly deep structure may introduce redundant propagation and weaken the discriminative power of node representations. Therefore, sensitivity analysis of the number of graph neural network layers helps to explain the rationality of the structural modeling depth chosen in this paper's algorithm. The experimental results are shown in Figure 2.

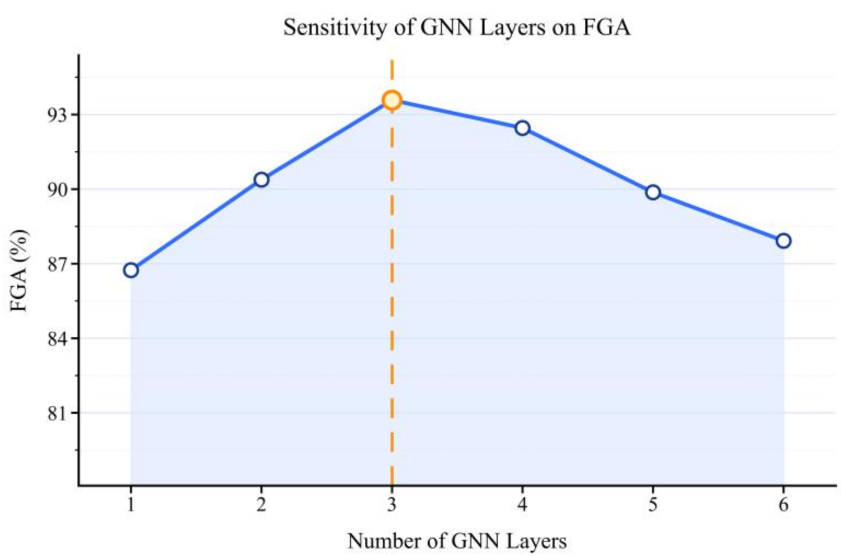

Figure 2. The impact of the number of layers in a graph neural network on FGA

Sensitivity results for graph neural network layers demonstrate that the proposed algorithm achieves more comprehensive process semantic representation and dependency modeling capabilities at a moderate structural depth. Shallow graph propagation layers struggle to cover cross-source field mappings, task dependencies, and rule constraint propagation relationships in enterprise-level ETL state graphs, resulting in insufficient awareness of the global process structure in node representations. As propagation depth increases further, excessive neighborhood information introduces redundant dependencies and semantic mixing, weakening the distinctiveness of key nodes and edges in process orchestration. The proposed algorithm strikes a good balance between structural modeling depth, semantic preservation capability, and process generation stability, indicating that relation-aware graph representation learning can provide effective structural support for automated orchestration processes and further improve business requirement coverage and process generation quality.

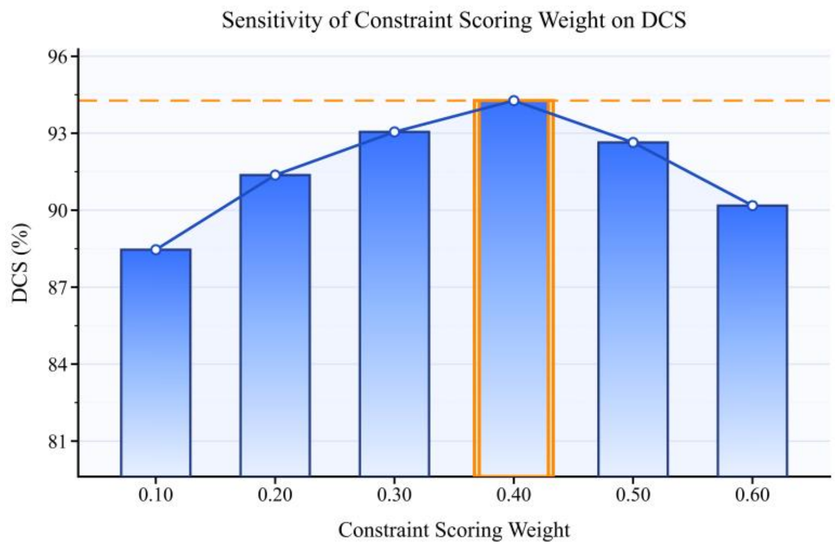

Figure 3. Impact of Constraint Scoring Weight Sensitivity on DCS

The constraint scoring weight sensitivity results demonstrate that the proposed algorithm exhibits good parameter adaptability during the reliable process selection stage. As shown in Figure 3, appropriate constraint scoring weights can strengthen the candidate ETL process's focus on field dependencies, task order, and structural consistency, making the generated results more stable in terms of data link organization and process dependency maintenance. When the constraint weights are set too low, the process selection process lacks sufficient constraint strength on structural constraints, which can easily weaken the dependency maintenance ability between data objects; when the constraint weights are set too high, the model may overemphasize the satisfaction of local constraints, thereby reducing the coordination between business semantics and overall process planning. The algorithm in this paper can achieve a good balance between structural constraints, semantic reasoning, and process executability, indicating that constraint consistency verification and multi-criteria scoring mechanisms can effectively support the reliable generation of enterprise-level ETL automatic orchestration.

5. Conclusion

This paper addresses the challenges of complex data relationships, variable business semantics, scattered rule constraints, and difficulties in unified modeling of process dependencies in enterprise-level ETL process auto-orchestration. It proposes an auto-orchestration method based on collaborative reasoning using graph neural networks and a large language model. This method unifies key objects such as data sources, fields, rules, and tasks into an ETL state graph. It uses relation-aware graph representation learning to characterize the structural dependencies and constraint propagation relationships between multi-source data, and leverages a large language model to perform task decomposition and process reasoning based on business requirements and rule semantics. Furthermore, a structural semantic joint context is used to guide candidate process generation, and constraint consistency verification and a multi-criteria scoring mechanism further improve the credibility and executability of the generated processes. This research advances the process orchestration problem in enterprise data engineering from manual rule configuration and static script generation to an automated stage driven by collaborative structural relationship modeling and semantic reasoning, providing a new technical approach for the intelligent construction of complex enterprise data pipelines.

In terms of overall performance, the proposed algorithm demonstrates good comprehensive performance in terms of process generation correctness, dependency consistency, rule coverage, and execution stability. Comparative experimental results demonstrate that graph-based relational modeling enhances the model's understanding of field mappings, task chains, and constraint relationships, while the semantic reasoning capabilities of the large language model further improve the quality of the transformation from business requirements to ETL steps. Compared to traditional methods relying on manual configuration, rule matching, or local process optimization, the proposed method can simultaneously address structural dependencies, semantic understanding, and process selection within a unified framework, making the generated ETL process more aligned with the actual needs of enterprise data governance and data warehouse construction. This method has positive implications for reducing enterprise data process development costs, improving cross-system data processing consistency, and enhancing the maintainability and reliability of data pipelines. It can also provide more stable data engineering support for intelligent data governance, automated data integration, data warehouse construction, and enterprise-level decision analysis platforms.

Future research can further extend the proposed method to more open and dynamic enterprise data environments. On one hand, finer-grained operational status feedback and online update mechanisms can be introduced, enabling the automatic orchestration model to continuously adjust when data patterns change, business rules are updated, and task execution anomalies occur, thereby improving the adaptability of the ETL process during long-term operation. On the other hand, it can further enhance the interpretability of process-generated results, enabling data engineers to more intuitively understand the basis for field mapping, the source of task dependencies, and the rule validation process, thereby improving the auditability and acceptability of automated orchestration systems in real enterprise environments. As enterprise data platforms continue to evolve towards intelligence, automation, and trustworthiness, the collaborative reasoning mechanism of graph neural networks and large language models is expected to be further applied to scenarios such as data quality repair, data lineage analysis, automated data governance, and cross-domain data integration, providing support for enterprises to build more efficient, reliable, and scalable data infrastructure.

References

[1] Y. Lu, "Policy-Aware Enterprise Risk Assessment through LLM-Based Cross-Source Representation and Reasoning," 2024.

[2] M. Zaharia, T. Das, H. Li, et al., "Discretized Streams: Fault-Tolerant Streaming Computation at Scale," Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, pp. 423-438, 2013.

[3] L. Liu, S. Hasegawa, S. K. Sampat, et al., "Autodw: automatic data wrangling leveraging large language models," in Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, 2024, pp. 2041-2052.

[4] H. Cui, "Task Graph Modeling and Constraint-Aware Scheduling for Reliable Large Language Model Agent Execution," 2024.

[5] Y. Xue, "Enhancing Trustworthiness of Retrieval-Augmented Large Language Models via Confidence Calibration and Selective Rejection," 2024.

[6] A. Remadi, K. El Hage, Y. Hobeika, et al., "To prompt or not to prompt: Navigating the use of large language models for integrating and modeling heterogeneous data," Data & Knowledge Engineering, vol. 152, p. 102313, 2024.

[7] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and S. Y. Philip, "A Comprehensive Survey on Graph Neural Networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4-24, 2021.

[8] G. Jaimovitch-López, C. Ferri, J. Hernández-Orallo, et al., "Can language models automate data wrangling?," Machine Learning, vol. 112, no. 6, pp. 2053-2082, 2023.

[9] J. Giovanelli, B. Bilalli, and A. Abello, "Data pre-processing pipeline generation for AutoETL," Information Systems, vol. 108, p. 101957, 2022.

[10] N. Konstantinou and N. W. Paton, "Feedback driven improvement of data preparation pipelines," Information Systems, vol. 92, p. 101480, 2020.

[11] L. Berti-Equille, "Learn2clean: Optimizing the sequence of tasks for web data preparation," in Proceedings of the World Wide Web Conference, 2019, pp. 2580-2586.

[12] R. Lourenço, J. Freire, and D. Shasha, "Bugdoc: A system for debugging computational pipelines," in Proceedings of the 2020 Association for Computing Machinery Special Interest Group on Management of Data International Conference on Management of Data, 2020, pp. 2733-2736.

[13] M. Mahdavi and Z. Abedjan, "Baran: Effective error correction via a unified context representation and transfer learning," Proceedings of the Very Large Data Bases Endowment, vol. 13, no. 12, pp. 1948-1961, 2020.

[14] S. Kandel, A. Paepcke, J. Hellerstein, et al., "Wrangler: Interactive visual specification of data transformation scripts," in Proceedings of the Association for Computing Machinery Special Interest Group on Computer-Human Interaction Conference on Human Factors in Computing Systems, 2011, pp. 3363-3372.